Intelligent Automation And Soft Computing, 2018
Copyright © 2018, TSI® Press
Vol. 24, no. 3, 593–602

The SLAM Algorithm for Multiple Robots based on Parameter estimation

MengYuan Chen[1 2]

[1]. Key Lab of Electric Drive and Control of Anhui Province, Anhui Polytechnic University, Wuhu 241000, China
[2]. Department of Precision Machinery and Precision Instrumentation, University of Science and Technology of China, Hefei 230027, China

ABSTRACT
With the increasing number of feature points of a map, the dimension of systematic observation is added gradually, which leads to the deviation of the volume points from the desired trajectory and significant errors on the state estimation. An Iterative Squared-Root Cubature Kalman Filter (ISR-CKF) algorithm proposed is aimed at improving the SR-CKF algorithm on the simultaneous localization and mapping (SLAM). By introducing the method of iterative updating, the sample points are re-determined by the estimated value and the square root factor, which keeps the distortion small in the highly nonlinear environment and improves the precision further. A robust tracking Square Root Cubature Kalman Filter algorithm (STF-SRCKF-SLAM) is proposed to solve the problem of reduced accuracy in the condition of state change on the SLAM. The algorithm is predicted according to the kinematic model and observation model of the mobile robot at first, and then the algorithm updates itself by spreading the square root of the error covariance matrix directly, which greatly reduces the computational complexity. At the same time, the time-varying fading factor is introduced in the process of forecasting and updating, and the corresponding weight of the data is adjusted in real time to improve the accuracy of multi-robot localization. The results of simulation shows that the algorithm can improve the accuracy of multi-robot pose effectively.

KEY WORDS: Iteration, Simultaneous localization and mapping, Sampling, Multi-mobile robot, Square Root Cubature Kalman Filter, Strong tracking

1 INTRODUCTION

SIMULTANEOUS Localization and Mapping (SLAM) can be expressed as follows: Via gathering information from the sensors, a multi-mobile robot is capable of creating the feature map in the unknown environment, and modifying the position and orientation continuously (Ullah, et.al. 2017) (LIU, et.al. 2011). It is a hot topic for mobile robots whether to motion autonomously in the field of SLAM.

The mainstream algorithm applied in the field of SLAM is the classical Extended Kalman Filter (EKF), which is proposed by Smith and Cheeseman, and its essence is first-order Taylor series expansion about the nonlinear system model and observation model, which is linearized by means of the standard Kalman Filter (Smith, et.al. 1987) (Smith, et.al. 1990). On the basis of EKF-SLAM, many scholars have put forward a new improved algorithm. Iterative Extended Kalman Filter (IEKF) proposed relies on the combination of environment feature iteration and EKF algorithm, which corrects the nonlinear error of the robot and improves the accuracy of localization (Qiang, et.al. 2013). An anti-interference EKF-SLAM algorithm, based on the detection of external interference by contrast and errors of expansion positioning, is put forward to improve the robustness of the algorithm (Tai, et.al. 2012). The higher-order term is ignored in the process of linearization, and so, it is not easy to achieve the desired results under circumstance of precisely positioning. The Jacobian matrix calculated in the linearization procedure increases the complexity of computation, which is not applicable to a broad environment as well. Multi-robot coordinated control is a hotspot of robot research in recent years, and there is a broad applying prospect in industrial, military, aerospace and so on (WANG, et.al. 2015) (ZHU, et.al. 2015).

CONTACT MengYuan Chen mychen@ahpu.edu.cn

Most of the robot systems require that robots can be aware of surroundings and capable of locating itself when exploring the environment. The multi-robots observe mutually and share information by exchanging messages, which reduces the dependency on the external environment and enhances the perception of a single robot, so it can obtain more accurate positioning information than a single robot does, which is named as an approach of cooperative localization for multiple robots. The co-localization study was originally focused on land-based robots, and later developed into underwater robots, parallel structures composed of homogeneous robots (Ling, et.al. 2007), and master-slave structures composed of heterogeneous robots (Bailey, et.al. 2011). In addition, transferring information from the superior quality sensor information to the inferior quality sensor information can obtain overall precision of the system to be best.

In recent years, the popular algorithm Unscented Kalman Filter (UKF) was proposed (Ouyang, et.al. 2014) (LIN, et.al. 2013). The sampling points with different weights (Sigma points) are chosen to be transferred with the non-linear function, and the statistical characteristics of the random variables are obtained by UT.

Under the framework of the Kalman Filter, UKF reduces the errors caused by the linearization of nonlinear equations and achieves the second-order positioning accuracy under the same conditions. In literature (Merwe, et.al. 2004), Merwe and Wan develop this theory effectively by deriving mathematically the square root of covariance matrix instead of transferring covariance matrix, which can be well applied to the SLAM (Zhao, et.al. 2011). In literature (WANG, et.al. 2014), based on the square root UKF, the sampling strategy of the Sigma point is changed and a single-row sampling algorithm is proposed, but UKF may lead to non-definite covariance matrices in the process of filtering, which affects filter performance.

In 2009, Arasaratnam and Haykin proposed the Cubature Kalman Filter (CKF), which provides a new method for state estimation of the nonlinear system. Under the rules of volume in CKF, a set of the corresponding weight of the volume point was selected into the non-linear function (BIAN, et.al. 2017) (ZHAO, et.al. 2017) (Arasaratnam, et.al. 2009), and statistical properties can be obtained by processing the weighted distribution of random variables. Compared with UKF, CKF also reduces the errors caused by the linearization of nonlinear equations and this algorithm has been recognized by more and more scholars for its low computational complexity, high numerical precision and strong filter stability, which makes it effective to solve various valuation problems. Square Root Cubature Kalman Filter (SR-CKF) is put forward, which can enhance the stability of system state (KANG, et.al. 2013). However, if the system state changes suddenly, SR-CKF positioning has a poor performance. A Strong Tracking Filter (STF) is proposed, and time-varying fading factor is introduced when recursively updating, which achieves a dynamic adjustment of the gain matrix, and the improved SLAM is still capable of highly tracking even if the system changes the state abruptly, but at the same time it greatly increased the computational complexity (WANG, et.al. 2013).

Therefore, Iterative Square Root Cubature Kalman Filter (ISR-CKF) is presented in this paper, which introduces Gauss-Newton iterative method on the basis of SR-CKF and draws the advantage of SR-CKF; transferring the matrix information with sampling the square root of the covariance to reduce the truncation error on SLAM, increasing the iterative process to facilitate the full use of measurement information and reducing the SR-CKF algorithm in a highly nonlinear environment of the system state estimation error. Compared with the previous UKF and SR-CKF, ISR-CKF on SLAM can improve the accuracy of the robot pose estimation effectively by simulation software MATLAB. In this paper, SR-CKF combined with the theory of strong tracking filter and Strong Tracking Filter Square Root Cubature Kalman Filter SLAM algorithm (STF-SRCKF-SLAM) is proposed, which is to ensure that the system is positive and a strong tracking filter can dynamically adjust the weight of the corresponding data to improve the accuracy of the system. Experiments show that the STF-SRCKF-SLAM algorithm has a good performance on estimation accuracy and it can keep strong tracking ability under the circumstance of the mutation state, which can be used for simultaneous localization and mapping in large areas.

2 THE DESCRIPTION OF SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM) BASED ON BAYESIAN ESTIMATION

THE essence of SLAM is to estimate the pose of the robot and feature map by the movement information inside the mobile robot and the observation information collected from the sensors.

Define the following variables at this time of K: $S_k = \left[S_k^r, M_k^i \right]^T$ represents the robot system time vector, of which S_k^r represents the robot pose information. $M_k^i = [\mathrm{m}_k^1, \mathrm{m}_k^2, \cdots, \mathrm{m}_k^i]^T$ signifies K-time map feature vector and n_s denotes the system state vector dimension. $U_{1:k} = \left[u_1, u_2, \cdots, u_k \right]^T$ embodies the kinematic information of the robot at the time of k and n_u symbolizes the vector control vector dimension of the robot. $Z_{1:k} = \left[z_1, z_2, \cdots, z_k \right]^T$ Represents the observation vector of the robot k and n_z represents the observation vector of the robot at the time of k.

The motion model of the robot is $S_k^r = f\left(S_{k-1}^r, u_k\right) + W_k$. The robot observation model is $z_k = h\left(S_k^r\right) + V_k$, of Which W_k obey the Gaussian distribution and V_k obey the Gaussian distribution (YANG, et.al. 2014).

SLAM can be expressed by

$$p\left(S_k^r, M_k^i \middle| Z_{1:k}, U_{1:k}\right), \tag{1}$$

Namely the known controlled quantity $U_{1:k}$ and observations $Z_{1:k}$ estimate the feature map M_k^i and the state vector S_k^r of the joint probability distribution at the moment. (1) can be obtained at each moment to meet the normal distribution of the optimal solution $\hat{S}_k$ and its covariance P_k, which is the key to SLAM.ISR-CKF proposed in this paper can estimate the pose of the robot and the feature map through the known information and its core is to calculate the conditional probability density by volume transformation (2) and realize Bayesian filtering under Gaussian domain.

$$\begin{aligned} I &= \int f(y) N\left(y; \mu, \textstyle\sum\right) dy \\ &\approx \frac{1}{2n_y} \sum f\left(\sqrt{\textstyle\sum}\, \xi_i + \mu\right) \end{aligned} \tag{2}$$

ξ_i is expressed as a set of perfectly ortho-symmetric volume points to each other, where ξ_i is $2\left(n_s + n_u\right)$ columns.

$$\xi_i = \sqrt{n_s + n_u} \left\{ \begin{bmatrix} 1 \\ 0 \\ \cdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \cdots \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ \cdots \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \cdots \\ 0 \\ 1 \end{bmatrix}, \cdots, \begin{bmatrix} -1 \\ 0 \\ \cdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ \cdots \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ \cdots \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \cdots \\ 0 \\ -1 \end{bmatrix} \right\} \tag{3}$$

Based on the principles of the Bayesian filtering, the ISR-CKF algorithm is divided into two steps, which are prediction and renewal to achieve prior probability distribution and posterior probability distribution at the time of k.

First, a prior probability distribution is obtained by using the posterior probability distribution at time k-1 and the quantity of motion control at time K:

$$p\left(S_k \middle| Z_{1:k-1}, U_{1:k}\right) = \int p\left(S_k^r \middle| S_{k-1}^r, U_{1:k}\right) \cdot p\left(S_k \middle| Z_{1:k-1}, U_{1:k-1}\right) dS_{k-1} \tag{4}$$

Second, the posteriori probability distribution is updated with the k-time observation and the prior probability distribution:

$$p\left(S_k \middle| Z_{1:k}, U_{1:k}\right) = \eta p\left(z_k \middle| S_k\right) \cdot p\left(S_k \middle| Z_{1:k-1}, U_{1:k}\right) \tag{5}$$

η is a constant

3 ITERATIVE SQUARE ROOT CUBATURE KALMAN FILTER ON SLAM (ISR-CKF-SLAM)

THE SR-CKF algorithm is based on the third-order spherical-phase volume rule in the framework of the Kalman Filter and the mean value of the non-linear function and the square root of the covariance are obtained by weighting the set of points. The covariance matrix square root (based on QR decomposition) instead of covariance matrix is used to solve the filter divergence, which is caused by rounding error. The specific algorithm in prediction process is described as follows:

(1) Forecast stage:

① Calculate the volume points:

$$x_{k-1}^j = L_{k-1}^A \xi_j + S_{k-1}^A \tag{6}$$

where j is the volume point number and its value is $1, 2, 3, \cdots 2\left(n_u + n_s\right)$ x_{k-1}^j contains information about position, feature points and motion control at the time of k-1. Referring from the information about the state S_{k-1} and the motion u_k, which is expanded to Gaussian noise variation, L_{k-1}^A and S_{k-1}^A can be obtained from (7)

$$S_{k-1}^A = \begin{bmatrix} S_{k-1} \\ u_k \end{bmatrix}, L_{k-1}^A = \begin{bmatrix} L_{k-1} & 0 \\ 0 & \sqrt{Q_k} \end{bmatrix} \tag{7}$$

② Calculate the priori estimates of each volume point and take volume point j as an example:

(8)

After each volume point has been propagated through the nonlinear motion equation, it can get the information about the position of the robot at K-1 time, the motion at K time, and predicts the position of the robot at K time.

③ Estimation on position state of the robot and predictions for square root factor

According to the volume transformation (2)

$$S_{k|k-1} = \frac{1}{2\left(n_s + n_u\right)} \sum_{j=1}^{2\left(n_s + n_u\right)} x_{k|k-1}^j \tag{9}$$

The square root factor $C_{k|k-1}$, which is for the update phase, can be obtained after the QR decomposition of the error vector.

$$A_{k|k-1} = \frac{1}{\sqrt{2\left(n_s + n_u\right)}} \begin{bmatrix} x_{k|k-1}^1 - S_{k|k-1} & x_{k|k-1}^2 - S_{k|k-1} & \cdots & x_{k|k-1}^{2\left(n_s + n_u\right)} - S_{k|k-1} \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} Q & R \end{bmatrix} = qr\left(A_{k|k-1}{}^{T}\right), C_{k|k-1} = R^{T} \tag{11}$$

Although the SR-CKF-SLAM algorithm can improve the positional accuracy and the system stability of the mobile robot, the dimension of the system observation will be added with the feature points on the map increasing gradually, which leads to the deviation of volume points from the ideal trajectory and therefore results in a large error in the state estimation. In this paper, the iterative method is used to reconstruct the sampling points in the update stage with the usage of the estimated value and the square root factor. The statistic properties of the system are obtained by volume transformation, and then the system is combined with the new observations estimated in the prediction phase to improve the system. The specific update phase algorithm is described as follows:

(2) Update phase

Take the feature point of number i as an example and suppose the iterative initial values are $S_{k|k-1}$ and $C_{k|k-1}$ respectively. The lth iterative robot position and square root factor are $S_{k|k-1}^{(l)}$, $C_{k|k-1}^{(l)}$. The observed value of time k is z_k^i and its observation model is

$$z_{k|k-1}^{i} = h\left(S_k^r\right) + V_k, \tag{12}$$

Calculate the iteration volume point:

$$D_{k-1}^{j,(l)} = C_{k|k-1}^{(l)} \xi_j + S_{k|k-1}^{(l)}, \tag{13}$$

② Calculate the lth iterative Kalman gain:

$$z_{k|k-1}^{i,j,(l)} = h\left(D_{k-1}^{i,j,(l)}\right), \tag{14}$$

$$\hat{z}_{k|k-1}^{i,(l)} = \frac{1}{2n_s} \sum_{j=1}^{2n_s} z_{k|k-1}^{i,j,(l)}, \tag{15}$$

$$B_{k|k-1}^{i,(l)} = \frac{1}{\sqrt{2n_s}} \begin{bmatrix} z_{k|k-1}^{1} - \hat{z}_{k|k-1}^{i,(l)} & z_{k|k-1}^{2} - \hat{z}_{k|k-1}^{i,(l)} & \cdots & z_{k|k-1}^{2(n_s+n_u)} - \hat{z}_{k|k-1}^{i,(l)} \end{bmatrix}, \tag{16}$$

$$\begin{bmatrix} Q & R \end{bmatrix} = qr\left(\begin{bmatrix} B_{k|k-1}^{i,(l)} & \sqrt{R_k} \end{bmatrix}^{T}\right) \quad P_{k|k-1}^{zz,(l)} = R^{T}, \tag{17}$$

$$P_{k|k-1}^{xz,(l)} = A_{k|k-1}^{(l)} B_{k|k-1}^{(l)\,T}, \tag{18}$$

Kalman gain:

$$W_k^{(l)} = P_{k|k-1}^{xz,(l)} \cdot \left(P_{k|k-1}^{zz,(l)T} \cdot P_{k|k-1}^{zz,(l)}\right)^{-1}, \tag{19}$$

③ The robot pose $S_{k|k-1}^{(l+1)}$ and the square root factor $C_{k|k-1}^{(l+1)}$ are calculated when iterating l+1 times.

$$S_{k|k-1}^{(l+1)} = S_{k|k-1} + W_k^{(l)} \left[z^k - h\left(S_{k|k-1}^{(l)}\right) - P_{k|k-1}^{xz,(l)T} \cdot B_{k|k-1}^{i,(l)\,-1} \left(S_{k|k-1} - S_{k|k-1}^{(l)}\right) \right], \tag{20}$$

$$C_{k|k-1}^{(l+1)} = C_{k|k-1} - W_k^{(l)} \cdot P_{k|k-1}^{zz,(l)} \cdot \left(W_k^{(l)}\right)^{T}, \tag{21}$$

④ Set the condition of stopping iteration

$$l = L_{\max}, \tag{22}$$

L_{max} is the maximum number of iterations, which is a fixed constant set in advance (22)

⑤ Stop iteration and update each data:

$$S_{k|k-1} = S_{k|k-1}^{(L_{\max})}, \tag{23}$$

$$C_{k|k-1} = C_{k|k-1}^{(L_{\max})}, \tag{24}$$

Update the position

(25)

Repeat equations (12) to (25) when multiple feature points are observed

4 AN IMPROVED MULTI-ROBOT COOPERATIVE LOCATION BASED ON SQUARE ROOT CUBATURE KALMAN FILTER

DURING the movement of the robot, the position of the robot R_j is observed by the robot R_i at time k, by means of obtaining the relative azimuth between them with the external sensors. As is shown in Fig. 1, θ_i and θ_j are the two robots' direction of movement respectively, and θ_{ij} is the relative azimuth angle of the robot R_a to the robot R_b, equation can be expressed as:

$$\phi_{ij} = \arctan\left[\frac{y_k^j - y_k^i}{x_k^j - x_k^i}\right] - \theta_k^i \tag{26}$$

The general form of observation model can be obtained:

$$z_k^{ab} = h(S_k^a, S_k^b) + V_k \tag{27}$$

where $z_k \in R^{n_z}$ is the observation matrix of the system with -dimensional vector at time k, and is the observation matrix whose variance is and the matrix is subject to the Gaussian distribution of $N(0, R_k)$.

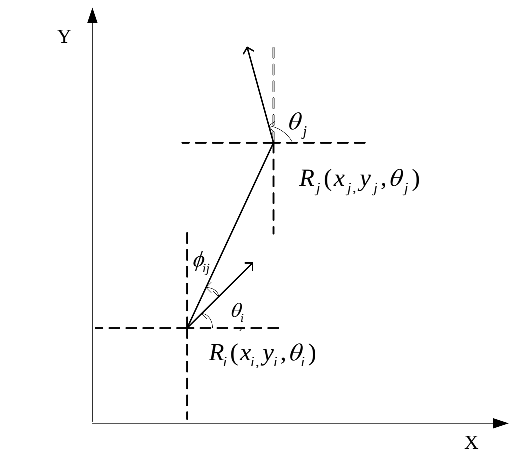


Figure 1. The Relative Observation between R_i and R_j.

The environment map is represented by a set of feature points and when the robot obtains the motion data, the relative observation obtained by the motion model is combined with the sensor to update the position of the robot and the square root factors on the entire queue. The multi-robot cooperative localization algorithm is described in detail with the robot R_i as an example due to the same process for updating the state of each robot at the time of k.

When a plurality of feature points is observed at the same time, it is necessary to calculate equations (28) and (29) and repeat the steps of Fig 2.

$$s_{k|k-1} = s_k, \tag{28}$$

$$C_{k|k-1} = C_k, \tag{29}$$

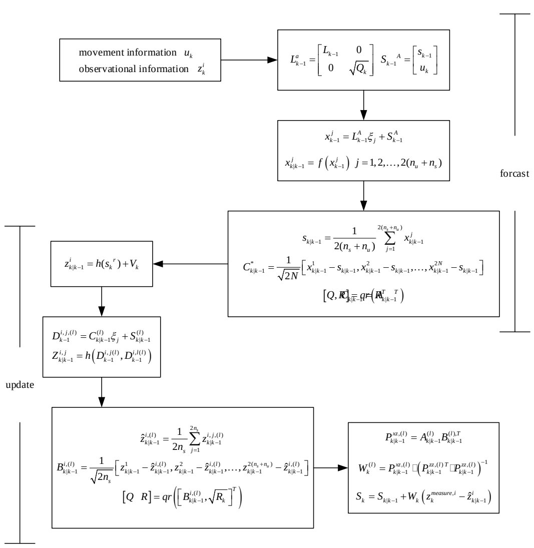


Figure 2. Block Diagram of the Multi-robot Cooperative Localization.

Although the SR-CKF-SLAM algorithm can improve the positional accuracy and the system stability of the mobile robot, the dimension of the system observation will be added with the feature points on the map increasing gradually, which leads to the deviation of volume points from the ideal trajectory and therefore results in a large error in the state estimation. In this paper, a Strong Tracking Square Root Cubature Kalman Filter SLAM (STF-SRCKF-SLAM) algorithm is proposed to solve the divergence of the numerical calculation when the state is abruptly changed. The algorithm proposed can also correct the corresponding estimation bias, guarantee the tracking performance and improve the pose accuracy.

5 STRONG TRACKING FILTER SQUARE ROOT CUBATURE KALMAN FILTER SLAM (STF-SRCKF-SLAM)

5.1 Strong Tracking Filtering Algorithm

AIMING at the problem of filter divergence and numerical instability caused by model uncertainty or system mutation, the strong tracking filter algorithm improves the stability and accuracy of the system in map building by real-time online adjustment, increases the weight of new data, and weakens the influence of old data precision.

The time-varying fading factor in the strong tracking filter can be calculated as follows:

$$V_k = \begin{cases} e_1 e_1^T, k = 1 \\ \dfrac{\rho V_{k-1} + e_k e_k^T}{1+\rho}, k \geq 2 \end{cases}, \tag{30}$$

where V_k is the residual covariance matrix, ρ is the forgetting factor, which is range from 0.95 to 0.98 e_k is the output residual sequence, which can be calculated by (5)

$$e_k = z_k - \hat{z}_{k|k-1}, \tag{31}$$

$$N_k = V_k - H_k Q_{k-1} H_k^T - \beta R_k, \tag{32}$$

$$M_k = H_k F_{k|k-1} P_{k-1|k-1} F_{k|k-1}^T H_k^T, \tag{33}$$

where $F_{k|k-1}$ and H_k are the first-order partial derivative matrix of the state equation and the measurement equation to the state variable. β is a weakening factor whose value can be determined according to system change.

Time-varying fading factor can be obtained from (34):

$$\lambda_k = \begin{cases} \lambda_0, \lambda_0 > 1, \\ 1, \lambda_0 \le 1, \end{cases} \lambda_0 = \frac{tr[N_k]}{tr[M_k]} \tag{34}$$

Where $tr(\cdot)$ indicates the trace of matrix.

5.2 Tracking Iterative Square Root Cubature Kalman Filtering Algorithm

According to the prediction stage and the update stage, the state prediction covariance matrix $P^*_{k|k-1}$, the output prediction covariance matrix $P^*_{zz,k|k-1}$, and the cross-covariance matrix $P^*_{xz,k|k-1}$ without introducing the fading factor. Equation can be obtained as follows:

$$\lambda_k = \begin{cases} \lambda_0, \lambda_0 > 1, \\ 1, \lambda_0 \le 1, \end{cases} \quad \lambda_0 = \frac{tr[N_k]}{tr[M_k]}$$

$$N_k = V_K - \left[P^*_{xz,k|k-1}\right]^T \left[P^*_{k|k-1}\right]^{-1} Q_{k-1} \left[P^{*,T}_{k|k-1}\right]^{-1} P^*_{xz,k|k-1} - \beta R_k$$

$$M_k = P^*_{zz,k|k-1} - V_k + N_k \tag{9}$$

From Fig.1, equations can be expressed as follows:

$$P^*_{k|k-1} = C^*_{k|k-1}\left(C^*_{k|k-1}\right)^T, \tag{35}$$

$$P^*_{xz,k|k-1} = C^*_{k|k-1}\left(Z^*_{k|k-1}\right)^T, \tag{36}$$

$$P^*_{zz,k|k-1} = Z^*_{k|k-1}\left(Z^*_{k|k-1}\right)^T, \tag{37}$$

Plug (10) - (12) into (9), equations can be obtained:

$$N_k = V_K - Z^*_{k|k-1}\left(C^*_{k|k-1}\right)^{-1} Q_{k-1}\left(C^*_{k|k-1}\right)^{-T}\left(Z^*_{k|k-1}\right)^T - \beta R_k, \tag{38}$$

$$M_k = Z^*_{k|k-1}\left(Z^*_{k|k-1}\right)^T - V_k + N_k, \tag{39}$$

$$\lambda_k = \begin{cases} \lambda_0, \lambda_0 > 1, \\ 1, \lambda_0 \le 1, \end{cases} \tag{40}$$

$$\lambda_0 = \frac{tr\left[V_K - Z^*_{k|k-1}\left(C^*_{k|k-1}\right)^{-1} Q_{k-1}\left(C^*_{k|k-1}\right)^{-T}\left(Z^*_{k|k-1}\right)^T - \beta R_k\right]}{tr\left[Z^*_{k|k-1}\left(Z^*_{k|k-1}\right)^T - V_k + N_k\right]}, \tag{41}$$

When the time-varying fading factor λ_k is introduced, the square root factor of the new state covariance prediction can be obtained as follows:

$$C_{k|k-1} = Tria\left(\left[\sqrt{\lambda_k} \cdot x_{k|k-1}, C_{Q,k-1}\right]\right) \tag{42}$$

Where $Tria(\cdot)$ denotes the triangular matrix obtained by QR decomposition, and $C_{Q,k-1}$ is the square root factor of the variance Q_k of the motion noise.

After introducing the fading factor, the square root $C_{k|k-1}$ of the prediction error variance matrix is calculated when updating Fig 1.

6 SIMULATION AND ANALYSIS OF THE ALGORITHM

The experiment to the accuracy of SLAM is applied in the Matlab7.0 software environment on the computer with the frequency of 3.4GHz, Core-i3 dual-core processor and 4G memory. On the open-source simulation platform provided by Australian scholar Tim Bailey, UKF-SLAM, SR-CKF-SLAM and ISR-CKF-SLAM are simulated respectively and they are also analyzed on the precision difference.

In the simulation experiment, the motion equation of the robot is:

$$m_k = \begin{bmatrix} m_{x,k} \\ m_{y,k} \\ m_{\theta,k} \end{bmatrix} = \begin{bmatrix} m_{x,k-1} + T \cdot v_k \cdot \cos(m_{\theta,k-1} + \alpha_k) \\ m_{y,k-1} + T \cdot v_k \cdot \sin(m_{\theta,k-1} + \alpha_k) \\ m_{\theta,k-1} + T \cdot v_k \cdot \frac{\sin(\alpha_k)}{L} \end{bmatrix} \tag{43}$$

where m_k represents the robot position, T represents internal sampling interval of the system, v_k represents the speed of the robot, α_k represents the rotation angle, and L represents the wheelbase.

The robot observation equation can be obtained in the simulation experiment:

$$z_k = \begin{bmatrix} l_k \\ \beta_k \end{bmatrix} = \begin{bmatrix} \sqrt{(m_{x,i} - m_{x,k})^2 + (m_{y,i} - m_{y,k})^2} \\ \arctan\left(\frac{m_{y,i} - m_{y,k}}{m_{x,i} - m_{x,k}}\right) - m_{\theta,k} \end{bmatrix} \tag{44}$$

l_k represents the distance between the robot and the observed feature points of the map, β_k represents the angle between the robot and the observed feature points of the map, and $(m_{x,i}, m_{y,i})$ is the position of the map's feature point observed by the robot.

The experimental hypothesis of the map environment is 250 × 200 outdoor environment with 17 identified path points and 35 map feature points. ISR-CKF-SLAM sets the number of iterations to five. The simulation results of the robot's trajectory are shown in Fig. 3, Fig. 4 and Fig. 5. The dotted line represents the ideal trajectory of the robot, the solid line represents the actual trajectory of the robot and the asterisk represents the map feature points. The simulation parameters can be seen in Table 1.

Table 1. **Simulation Parameters.**

Simulation parameters	Value	Simulation parameters	Value
Speed of the robot	3m/s	wheel gauge	4m
Maximum steering angle	±30º	Control noise from speed	0.3m/s
Maximum steering angular speed	±20º	Observation noise from distance	0.1m

6.1 Simulation Algorithm of the Iterative Square Root Cubature Kalman Filter

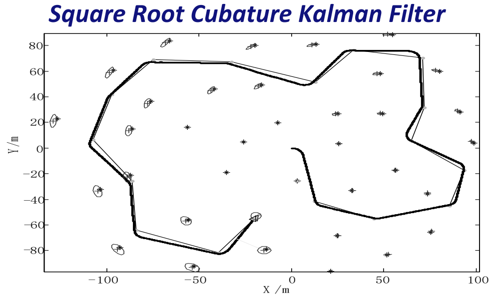


Figure 3. **UKF-SLAM.**

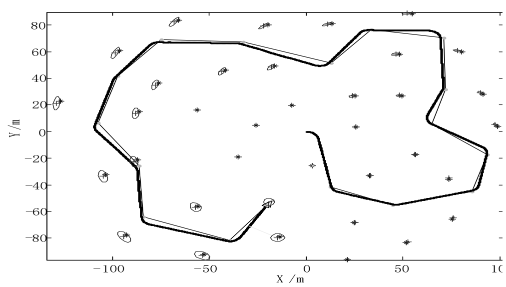


Figure 4. **SR-CKF-SLAM.**

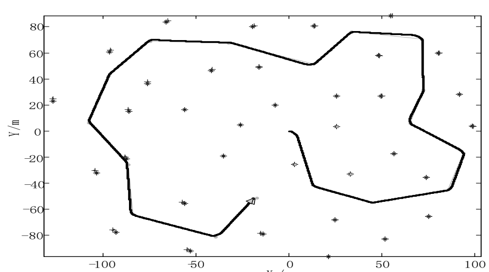
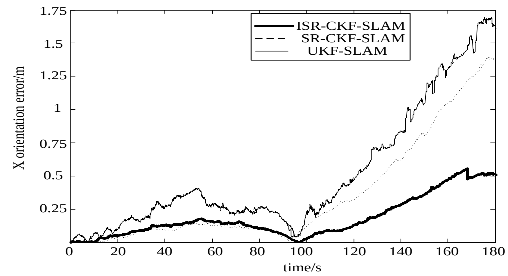


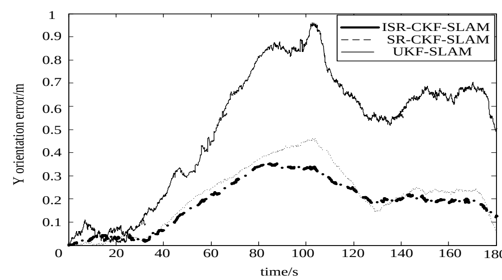
Figure 5. **ISR-CKF-SLAM.**

It is estimated that the whole path of the robot will take 240 seconds, and the average value will be selected after several experiments. In this paper, the first 180s of the simulation results are chosen to analyze.

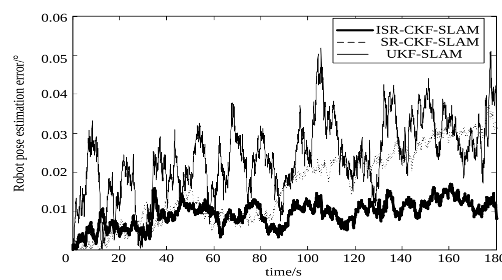
In the initial stage of travel, the running trajectories deviate little from the ideal trajectories of the three algorithms. However, with the increasing number of the feature points, the trajectories of the three algorithms have different degrees of deviation from the ideal trajectories, but the ISRs of the three algorithms are different from the ideal trajectories. Compared with UKF-SLAM and SR-CKF-SLAM, ISR-CKF-SLAM is more suitable for the ideal trajectory, which means the pose estimation is more accurate.



(a) Contrast in the X Direction's Error



(b) Contrast in the Y Direction's Error



(c) Contrast in the Angle Error

Figure 6. **Comparison of Three Algorithms for Estimation Errors.**

From Fig 6, it is shown that the evaluated error of ISR-CKF-SLAM in X direction, Y direction and angle is always within a stable range. Compared with UKF-SLAM and SR-CKF-SLAM, ISR-CKF-SLAM maintains a high accuracy.

Performance differences of these three kinds of algorithms can be expressed by statistical data.

Compared with SR-CKF-SLAM, the error in X direction of ISR-CKF-SLAM is reduced by 46.8%, the error in Y direction of ISR-CKF-SLAM is reduced by 13.2% and the error in angle is reduced by 46.6% from the data in Table 2.

Table 2. **Comparison in the Statistical Error of the Algorithm.**

SLAM algorithm	The average of the absolute value in the X direction's evaluated error /m	The average of the absolute value in the Y direction's evaluated error /m	The average of the absolute value in the angle error /rad
UKF	0.5773	0.5621	0.0348
SR-CKF-SLAM	0.4372	0.2436	0.0234
ISR-CKF-SLAM	0.2323	0.2113	0.0125

6.2 Simulation of Multi-robot Cooperative Localization on the Strong Tracking Filter Square Root Cubature Kalman Filter

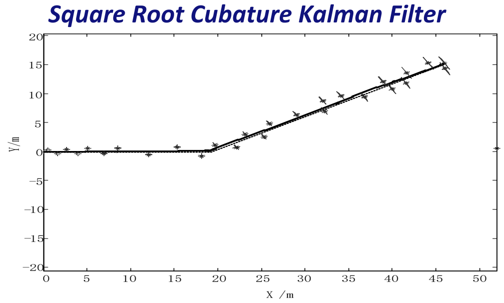


Figure 7. **STF-SRCKF-SLAM on a Single-robot.**

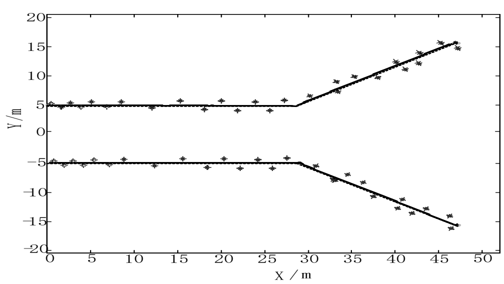


Figure 8. **STF-SRCKF-SLAM on a Multi-robot.**

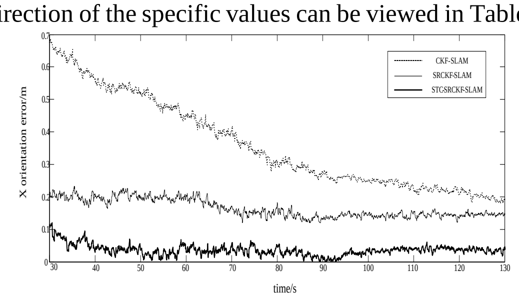
The proposed STF-SRCKF-SLAM algorithm is simulated in a single-robot and a multi-robot cooperative environment and is compared with the average of 30 experimental results of UKF-SLAM and SR-CKF-SLAM, the simulation results of the robot's trajectory are shown in Fig. 7 and Fig. 8, in which "dotted line" represents the ideal trajectory of the robot, "solid line" represents the actual trajectory of the robot, "*" indicates the map feature point and "+"indicates the position of the feature points.

Fig. 7 and Fig. 8 correspond to the experimental results of STF-SRCKF-SLAM on single-robot and multi-robot respectively. Compared with STF-SRCKF-SLAM on a multi-robot, STF-SRCKF-SLAM on a single-robot has a better performance on the actual route than the ideal route, which means that the accuracy of location estimation is higher and the map created is more accurate.

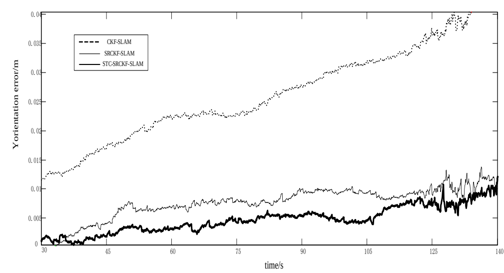
6.2.1 Error Analysis

The comparison curves of X-axis, Y-axis and angle estimation errors are shown in the same environment when a multi-robot is mapping on the CKF-SLAM, SRCKF-SLAM and STF-SRCKF- SLAM, where t represents the experimental time.

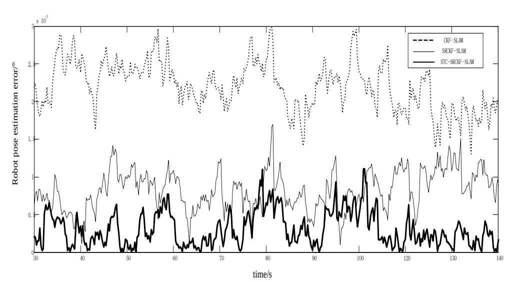
As is shown in Fig. 9, the best stability of the three algorithms is the STC-SRCKF-SLAM algorithm, then the SRCKF-SLAM algorithm, and the CKF-SLAM algorithm has the worst performance of them. The error of the three algorithms in the X-axis, Y-axis and angular direction of the specific values can be viewed in Table 2



(a) Error Comparison in the X Direction



(b) Error Comparison in the Y Direction



(c) Error Comparison on the Angle

Figure 9. **Comparison of the Three Algorithms for Estimation Errors.**

Table 3. Algorithm error

SLAM algorithm	The X-direction error varies most value /m	The X-direction error varies most value /m	Angle direction error maximum /°
STF-SRCKF	0.13	0.01	0.013
SRCKF	0.18	0.015	0.017
CKF	0.43	0.04	0.038

From Table 3, the error of the CKF-SLAM is the biggest, and the error of SRCKF-SLAM is smaller than that of the CKF-SLAM, because the square root factor is introduced to guarantee the semi-positive definite of the covariance matrix, which can restrain the truncation error of the computer. The fading factor is introduced to adjust the gain online on the STF-SRCKF-SLAM and the effect of the algorithm can be seen in Fig.9 that the accuracy of the algorithm is high and the error is small, and it is not easy to generate divergence of numerical filtering, which verifies validity and accuracy of the STF-SRCKF- SLAM algorithm.

6.2.2 Root Mean Square Error and Run Time Analysis.

Root mean square error of the formula is;

$$RMSE = \sqrt{\frac{1}{T}\sum_{k=1}^{T}\left[\left(x_k - \hat{x}_k\right)^2 + \left(y_k - \hat{y}_k\right)^2\right]}, \quad (45)$$

where T is the running time, and $\left(x_k, y_k\right)$ and $\left(\hat{x}_k, \hat{y}_k\right)$ are the actual position and the estimated position of the robot respectively at time k.

The RMSE and run-time of the three SLAM algorithms are showed in Table 4, and it can be seen from this table that the RMSE value of the STC-SRCKF-SLAM algorithm is the smallest, the estimation accuracy is the highest, and meanwhile the running time of STC-SRCKF-SLAM is the shortest.

Table 4. Experimental Statistics of the Three SLAM Algorithms.

SLAM algorithm	Root mean square error /m	operation hours /s
STC-SRCKF-SLAM	5.5880	159.5438
SR-CKF-SLAM	6.5906	161.6320
CKF-SLAM	7.7630	163.7232

7 CONCLUSION

IN this paper, an Iterative Squared Root Cubature Kalman Filter SLAM (ISR-CKF-SLAM) is proposed. The SR-CKF algorithm is improved by the Gauss-Newton iterative method. In this method, the latest updated information is efficiently used, which reduces the influence of initial error and linearization error on state estimation, and makes the posterior probability distribution closer to the true value. Compared with the UKF-SLAM and SR-CKF-SLAM, the ISR-CKF-SLAM is more accurate and more stable with the MATLAB simulation, which verifies the validity of the ISR-CKF-SLAM algorithm. A cooperative localization approach for multiple robots on the Strong Tracking Filter Square Root Cubature Kalman Filter SLAM proposed combines the advantages of the SRCKF and STF. On one hand, the integration of square root strategy can ensure the symmetry of the covariance matrix and effectively reduce the divergence caused by the system mutation filter. On the other hand, the weight of the corresponding data can be adjusted adaptively, and the accuracy of the system state estimation can be improved effectively by introducing the fading factor that can adjust the filter gain matrix in real time. It is the key problem to improve the real-time performance of the algorithm and reduce the influence of sensor observation error on the accuracy of the SLAM algorithm in large environment.

8 ACKNOWLEDGMENTS

This work was supported by 2018 Natural Science Foundation of Anhui, China (1808085QF215), 2018 Foundation for Distinguished Young Talents in Higher Education of Anhui, China (gxyqZD2018050) and Anhui Key Research and Development Programs (Foreign Scientific and Technological Cooperation, 1804b06020375).

9 REFERENCES

Arasaratnam, I., & Haykin, S. (2009). *Cubature kalman filters.* IEEE Transactions on Automatic Control, 54(6), 1254-1269.

Bailey, T., Bryson, M., Mu, H., & Vial, J. (2011). *Decentralised cooperative localisation for heterogeneous teams of mobile robots.* IEEE International Conference on Robotics and Automation (Vol.30, pp.2859-2865). IEEE.

Bian, M., Wang, J., Liu, W., & Qiu, K. (2017). *Robust and reliable estimation via recursive nonlinear dynamic data reconciliation based on cubature* kalman filter. Cluster Computing(6), 1-11.

Kang, Y., Song, Y., Song, Y., Yan, D., & Danyong, L. I. (2013). *Square-root cubature kalman filter and its application to SLAM of an mobile robot.* Robot, 35(2), 186.

Lin, W. (2013). *Design and implementation of control system for two-wheeled self-balancing robot.* Journal of Electronic Measurement & Instrument, 27(8), 750-759.

Ling, W., Liu, Y. H., Wan, J. W., & Shao, J. X. (2007). *Multi-robot cooperative localization based on relative bearing.* Chinese Journal of Sensors & Actuators, 20(4), 794-799.

Liu, D., Liu, G., & Yu, M. (2011). *Method of particle filter localization of robot based on* *heterogeneous sensor information fusion*. Journal of Electronic Measurement & Instrument, 25(1), 38-43.

Merwe, R. V. D., Wan, E. A., & Julier, S. I. (2004). *Sigma-point kalman filters for nonlinear estimation and sensor-fusion–applications to integrated navigation*.

Ouyang, G., & Zhou, J. (2014). *High accuracy ballistic filtering algorithm based on the kalman filter*. Electronic Measurement Technology.

Qiang, M., & Zhang, W. (2013). *Application of iterated extended kalman filtering in mobile robot localization*. Application of Electronic Technique.

Smith, R., Self, M., & Cheeseman, P. (1990). *Estimating Uncertain Spatial Relationships in Robotics*. *Autonomous robot vehicles*. Springer-Verlag New York, Inc.

Smith, R. C., & Cheeseman, P. (1987). *On the representation and estimation of spatial uncertainty*. International Journal of Robotics Research, 5, 56-68.

Tai-Zhi, L. V. (2012). *A novel ekf-slam algorithm against outlier disturbance*. Computer Engineering, 38(21), 1-4.

Ullah, J., & Jaffar, M. A. (2017). *Object and motion cues based collaborative approach for human activity localization and recognition in unconstrained videos*. Cluster Computing (2), 1-12.

Wang, B., Li, Y., Qi, G., & Sheng, A. (2015). *Bearings-only cooperative localization algorithm of multi-mobile robots based on square-root unscented kalman filter*. Nanjing LI Gong Daxue Xuebao/journal of Nanjing University of Science & Technology, 39(4), 440-446.

Wang, G., Chen, Y., & Chen, M. (2014). *Ukf-slam algorithm adopting square root and single row sampling method based on proportion and mini-skewness*. Computer Engineering & Applications.

Wang, H., Fu, G., Li, J., Li, C., & University, H. E. (2013). *Strong tracking ckf based slam method for unmanned underwater vehicle*. Chinese Journal of Scientific Instrument, 34(11), 2542-2550.

Zhao, L., Yan, X., & Hao, Y. (2011). *Slam algorithm based on iterated square unscented kalman filter*. Journal of Projectiles Rockets Missiles & Guidance, 31(1), 1906-1911.

Zhao, W., Li, H., Zou, L., & Huang, W. (2017). *Nonlinear unknown input observer based on singular value decomposition aided reduced dimension cubature kalman filter* Mathematical Problems in Engineering, 2017(3), 1-13.

Zhu, Q., Jia, W., Zhang, P., & Ying, C. (2015). *Research on mobile robot localization based on gaussian moment improved surf algorithm*. Chinese Journal of Scientific Instrument.

10 NOTES ON CONTRIBUTORS

Chen Mengyuan was born in 1984, and is an associate professor at the Anhui Polytechnic University. His main research filters include optimization of sensor information fusion and so on.